*Krishtechtrainings*

*+918595014719*

*https://krishtechtrainings.com/*

* What Is Dockerfile: Everything You Need to Know

Docker tools are a vital part of the management of most global companies. Docker tool runs an application with a high level of abstraction and security. Hence, many companies are extensively adopting the tool to achieve high network availability, service continuity, and service provision with high scalability.

AGENDA:

* What is Docker?
* What is a Dockerfile?
* List of Docker Commands for Creating a Dockerfile
* How to Create a Docker Image Using a Dockerfile?

## **What is Docker?**

Docker is a configuration management tool that is used to automate the deployment of software in lightweight containers. These containers help applications to work efficiently in different environments.

ocker container is a software package that has all the dependencies required to run an application.

Features of Docker:

* Easy and faster configuration
* Application isolation
* Security management
* High productivity
* High scalability
* Infrastructure independent

### **Docker Image:**

[A Docker](https://www.simplilearn.com/tutorials/docker-tutorial/what-is-docker?source) Image is a read-only file with a bunch of instructions. When these instructions are executed, it creates a Docker container.

### **Dockerfile:**

Dockerfile is a simple text file that consists of instructions to build Docker images.

Mentioned below is the syntax of a Dockerfile:

Syntax

# comments

command argument argument1...

Example

# Print "Get Certified. Get Ahead"

Run echo "Get Certified. Get Ahead"

Now, let's have a look at how to build a Docker image using a dockerfile.

## **List of Docker Commands for Creating a Dockerfile**

Before we create our first Dockerfile, it is important to understand what makes up the file.

Dockerfile consists of specific commands that guide you on how to build a specific Docker image.

The specific commands you can use in a dockerfile are:

FROM, PULL, RUN, and CMD

* FROM - Creates a layer from the ubuntu:18.04
* PULL - Adds files from your Docker repository
* RUN - Builds your container
* CMD - Specifies what command to run within the container

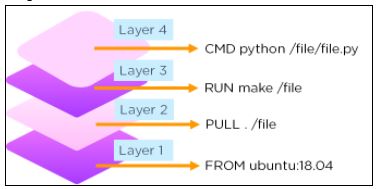
Mentioned below is an example of the dockerfile with the important commands

FROM ubuntu:18.04

PULL. /file

RUN make /file

CMD python /file/file.py



Moving forward, let’s go through some of the most common Docker commands used while creating dockerfiles. Along with the syntax, we are explaining the commands with examples, so you can start experimenting with them right away.

* ENTRYPOINT allows specifying a command along with the parameters

Syntax

ENTRYPOINT application "arg, arg1".

Example

ENTRYPOINT echo "Hello, $name".

* ADD command helps in copying data into a Docker image

Syntax

ADD /[source]/[destination]

Example

ADD /root\_folder/test\_folder

* ENV provides default values for variables that can be accessed within the container

Syntax

ENV key value

Example

ENV value\_1

* MAINTAINER declares the author field of the images

Syntax

MAINTAINER [name]

Example

MAINTAINER author\_name

## **How to Build a Docker Image and Docker Container Using Dockerfile?**

First of all, you should create a directory in order to store all the Docker images you build.

* Now, we will create a directory named ‘simplidocker’ with the command:

mkdir simplidocker

* Move Docker image into that directory and create a new empty file (Dockerfile) in it:

cd simplidocker

touch Dockerfile

* Open the file with the editor. In this example, we opened the file using vi:

vi Dockerfile

* Then, add the following content:

FROM ubuntu

MAINTAINER simpli

RUN apt-get update

CMD ["echo", "Welcome to Simplilearn"]

* 5.Save and exit the file.

## **Build a Docker Image with Dockerfile**

* docker build [OPTIONS] PATH | URL | -
* Now, let’s build a basic image using a Dockerfile:
* docker build [location of your dockerfile]
* Now, by adding -t flag, the new image can be tagged with a name:
* **docker build -t simpli\_image**

Once the Docker image is created, you can verify by executing the command:

docker images

The output should show simpli\_docker available in the repository.



## **Create a New Container**

Now, create a Docker container from the Docker image we created in the previous step.

Let’s name the container “simplilearn” and create it with the command:

docker run --name simplilearn simpli\_docker